# SPARTAN CONTROLS LTD.

## USDATA - FISHER ROC DRIVER
## USER'S MANUAL
## NT - 32 BITS

**SUPPORTS ECS**

**Version 1.0.1**
**April 1999**

# IMPORTANT NOTICE

Spartan Controls Ltd. (SCL) reserves the right to revise and improve the software and documentation at any time and without notice.

SCL makes no warranties, either expressed or implied, regarding the Fisher ROC Driver computer software package or its merchantability or fitness for any particular purpose.

SCL shall not be liable for technical or editorial omissions made herein; nor shall SCL be responsible for incidental or consequential damages resulting from the proper or improper use of these materials nor from the furnishing and performance of this material. The information contained herein does not represent a commitment on the part of SCL.



**Spartan Controls Ltd.**
305 - 27 Street S.E.
Calgary, Alberta
T2A 7V2

Phone:(403) 207-0700
Fax: (403) 207-0874

**DEFINITIONS**
"Spartan Controls" and "SCL" mean an Alberta Limited Company with respect to all Programs acquired by Customer.
"Program" means the SCL software product (executable program, documentation and/or data).
"Customer" means the person or organization in whose name the program was registered during software authorization.
"Single Machine" means the computer upon which the program will be installed and used, as identified in Customer's Purchase Order or as acknowledged during software authorization.

**LICENSE**
Customer may
(a) use the Program on a Single Machine at any one time;

(b) copy the Program only as reasonably necessary to support the authorized use, modification and merger. Each copy of the Program made by Customer must include a reproduction of any copyright notice or restrictive rights legend appearing in or on the copy of the Program as received from SCL.

Customer may not
(a) use the Program on more than one machine at any one time or in a network environment (except as authorized in a separate supplemental licensing agreement);

(b) transfer the Program to any person or organization outside of Customer or the corporation of which Customer is a part without the prior written consent of SCL;

(c) export or re-export, directly or indirectly, the Program, any associated documentation or the direct product thereof, to any country to which such export or re-export is restricted by law or regulation of Canada or any foreign government having jurisdiction without the prior authorization, if required, of the Ministry of External Affairs, Ottawa, Ontario, Canada and the corresponding agency of such foreign government or

(d) reverse compile or disassemble the Program for any purpose.

The design of SCL software products permits the Program to be downloaded or shared with additional machines for local execution. A separate license shall be required for each such machine with which the Program may be used or a supplemental agreement granting a network or site license shall be required for the entire configuration or installation within which the Program may be shared.

Title to the Program and all copies thereof, but not the media on which the Program or copies may reside, shall be and remain with SCL or others from whom SCL has obtained a licensing right.

Customer shall pay when due all property taxes that may now or hereafter be imposed, levied or assessed with respect to the possession or use of the Program or this license and shall file all reports required in connection with such taxes.

THE PROGRAM MAY NOT BE USED, COPIED, MODIFIED, MERGED OR TRANSFERRED TO ANOTHER EXCEPT AS EXPRESSLY PERMITTED BY THESE TERMS AND CONDITIONS.

UPON TRANSFER OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM, THE LICENSE GRANTED HEREIN IS AUTOMATICALLY TERMINATED.

**TERM**
The license granted herein is effective upon delivery of the Product to the Customer and shall remain in effect until terminated as provided herein. The license may be terminated by Customer at any time upon written notice to SCL. The license may be terminated by SCL or any third party from whom SCL may have obtained a respective licensing right if Customer fails to comply with any term or condition, including timely payment of licensing fees and such failure is not remedied within thirty (30) days after notice thereof from SCL or such third party. Upon termination by either party, Customer shall return to SCL the Program and all associated documentation, together with all copies in any form.

**LIMITED WARRANTY**
SCL warrants that the media on which the Program is furnished and the encoding of the Program on the media will be free from defects in materials and workmanship for a period of three (3) months from the date of shipment. If any such media or encoding proves defective during the warranty period, SCL will provide a replacement in exchange for the defective medium. Except as to the media on which the Program is furnished, the Program is provided "as is" without warranty of any kind, either express or implied. SCL does not warrant that the functions contained in the Program will meet Customer's requirements or that operation of the Program will be uninterrupted or error free.

THIS WARRANTY IS GIVEN BY SCL WITH RESPECT TO THE PROGRAM IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. SCL AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OF FITNESS FOR A PARTICULAR PURPOSE. RESPONSIBILITY TO REPLACE DEFECTIVE MEDIA OR REFUND CUSTOMER'S PAYMENT IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY.

**LIMITATION OF LIABILITY**
IN NO EVENT SHALL SCL OR OTHERS FROM WHOM SCL HAS OBTAINED A LICENSING RIGHT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL or CONSEQUENTIAL DAMAGES ARISING OUT OF OR CONNECTED WITH CUSTOMER'S POSSESSION OR USE OF THE PROGRAM, EVEN IF SCL OR SUCH OTHERS HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

**THIRD PARTY DISCLAIMER**
Except as expressly agreed otherwise, third parties from whom SCL may have obtained a license right do not warrant the Program, do not assume any liability with respect to its use and do not undertake to furnish any support or information relating thereto.

**U.S. GOVERNMENT RESTRICTED RIGHTS**
Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraph (a) through (d) of the Commercial Computer-Restricted Rights clause at FAR 52.227-19 when applicable or in subparagraph (C) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7013

**GENERAL**
This agreement contains the entire agreement between the parties with respect to the use, reproduction and transfer of the Program. Neither this Agreement nor the license granted herein is assignable or transferable by Customer without the prior written consent of SCL. This Agreement and the license granted herein shall be governed by the laws of the Province of Alberta, Canada.

**Spartan Controls Ltd.**
**305 - 27 Street S.E.**
**Calgary, Alberta T2A 7V2**
**Phone: (403) 207-0700 * Fax: (403) 207-0874**

This page intentionally left blank.

**Spartan Controls Ltd.**
**305 - 27 Street S.E.**
**Calgary, Alberta T2A 7V2**
**Phone: (403) 207-0700 * Fax: (403) 207-0874**

# TABLE OF CONTENTS

# Document Revision History

| Version # | Description | Date |
|---|---|---|
| 1.0.1 | Applies to Versions 1.06, 1.08.1, 1.10.1, 2.30.1 | April, 1999 |
| | | |
| | | |
| | | |

# Section 1

# INTRODUCTION

# 1. Introduction

This manual is designed to provide an overview of operations for the USDATA -Fisher ROC Driver. It will act as a *supplement* to other supplied documentation. It is assumed that the operator has at least basic knowledge of PC functions in the WINDOWS and DOS environments.

## 1.1 The I/O Interface Module.

**An Input/Output interface module is the communications link that lets FactoryLink WINDOWS NT talk with any of the Fisher Remote Operations Controllers (306, 312, 364 and the new ROC 407).** It links physically and electronically to the ROC. You must know how you want to define your data and how the FactoryLink configuration manager defines data ***before*** you configure the I/O interface module. Throughout this manual we refer to the ROC protocol in Fisher publication form A4199.

**Each ROC has a particular number of *Soft Points*.** A Soft Point is a reserved block of memory that stores information from a program. The ROC has up to 32 reserved blocks of memory. Each of these 32 Soft Points has 20 Data Points inside it. Each Data Point is a location for a value. These values helps store the information compiled by the ROC's program. Ultimately, the Host computer accesses the information of the program from the Soft Points.

There are several other terms you should know. A ***Parameter*** describes the limits of a given field, address or location. For example, an Analog Input has many parameters, including Engineering Units, Low AI, Hi AI. ***Implemented*** means active. An implemented point is a point currently in use or ready for use. A ***Field Point*** is a final end device and its characteristics. A Field Point usually monitors and/or generates information in the field.

**You must define Data Points in 4 separate locations for everything to work properly.**

1. **You must define the ROC Field Point.** You should know the ROC's configuration or have it on hand. This manual does not include the configuration details. You will find them in the *Fisher GV101 Configuration Software User Manual*, Form A4194.

2. **You must define your Communications system to receive common driver information.** Common driver information includes serial port name, baud rate and response time-outs. Each piece connects to an individual communications channel.
3. **You must define the Control Points of the ROC station.** Each active station's configuration differs. The fields in this part of the database define the

common points between the individual stations. These points include station name, ROC address, ROC group, communication status flags, etc.

4. **You must use the ROC Station Read/Write Information to define the individual Data Points.** You must make 1 separate database entry for each point. This entry consists of a tag name, point type, point number, parameter number, etc.

**The Spartan Controls Ltd. ROC Driver uses a standard serial asynchronous port (RS232C).** The communication channel is always open. The first to "talk" (transmit) is the first to get “heard” (received).

**The standard serial asynchronous port attaches to a Personal Computer (PC).** The driver does not support smart communication boards like the IBM Arctic Card. The ROC driver assumes a Full Duplex circuit is available, although it may actually use Semi- or Half-duplex circuits. As a result, the driver continuously listens for received data. This discrepancy between Full and Half Duplex circuitry does not affect communication. This manual assumes all communication channel equipment (i.e. modems, radios) works properly.

This page intentionally left blank.

# Section 2

# INSTALLING THE ROC DRIVER

# 2. Installing the ROC Driver

## 2.1 INSTALL

**The INSTALL command installs the drivers and associated files in the specified directories.** This command installs between 4 and 6 drivers. You can configure whichever driver you want, as they are all the identical. Follow the steps in **Sections 2.2** and **2.3** below. Note that the INSTALL command does not configure FactoryLink for you.

**Place the driver diskette in the appropriate floppy drive of the PC.**

**From the prompt, type: <INSTALL><Enter>**

Example: If the diskette is in A drive, change directory to A by typing **< cd a:\ >**. Then type: "INSTALL".

You will see a screen similar to the one below. Follow the prompts. You can abort the installation factor at any time by pressing one of these key sequences: **<Ctrl-c>** or **<Ctrl-Break>**.

```
ROC 364 driver installation (1.0) for Windows NT FactoryLink.
Six drivers will be installed, ROC 1, ROC 2, ROC 3, ROC 4, ROC 5 and ROC 6.
Input (control c) or (control break) to abort at any time.

An authorization number will now be requested, with only three tries allowed.

Spartan Controls Installation Authorization No.? xxxxxxxxxx
Your Company Name (3 to 29 chars.)? xxxxxxxxxxx

File C:\FLNT\ac\r1__.ac exists, overwrite (y,n)? y
```

**For each installation, you need an authorization number from Spartan Controls Ltd.** You can get it by telephoning **(403) 207-0700**. Ask for the **SCADA group**. To avoid delays in the installation process, you should get the authorization number before starting. As it says on the screen, you have only 3 chances to enter the correct authorization number. If you fail all 3 times, you will return automatically to the Windows NT prompt.

At the bottom of the screen you see a line asking if you want to overwrite an existing file. Type "**y**" or "**n**". Overwriting existing files while installing the drivers is normally OK.

## 2.2 Editing with the Configuration Manager Utility.

**You can edit the "*System Configuration*" table with the Configuration Manager Utility.** If you are in Windows NT, open the *FactoryLink* program group. Click on the *FactoryLink*

*Configuration Manager* icon. If you are at a dos prompt, type **<set FLAP = c:\flapp>** and press **<return>**. This sets the FL environment variable. Now type **<flcm>**. This will access to the FactoryLink Configuration Manager. ***System Configuration*** is listed in the ***Main Menu,*** as indicated below.

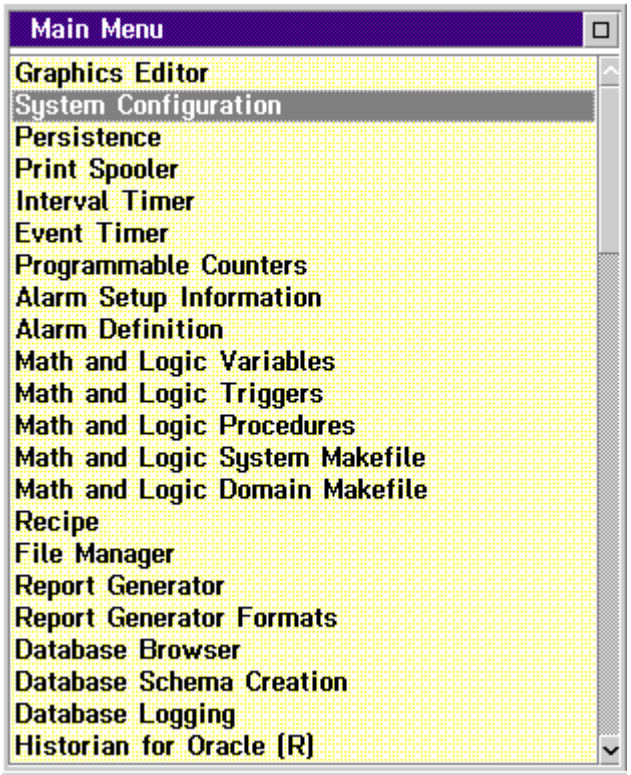


**Figure 2-1: Main Menu of the Configuration Manager Utility**

Type the number of the ROC you want to install (1 to 6). The drivers are listed as **1 ROC, 2 ROC,** etc.

In the shared domain, copy the last configured row in the System Configuration Table. Paste the row into the blank line directly after it, creating a duplicate. (To copy and paste, use the **Edit** pull-down menu).

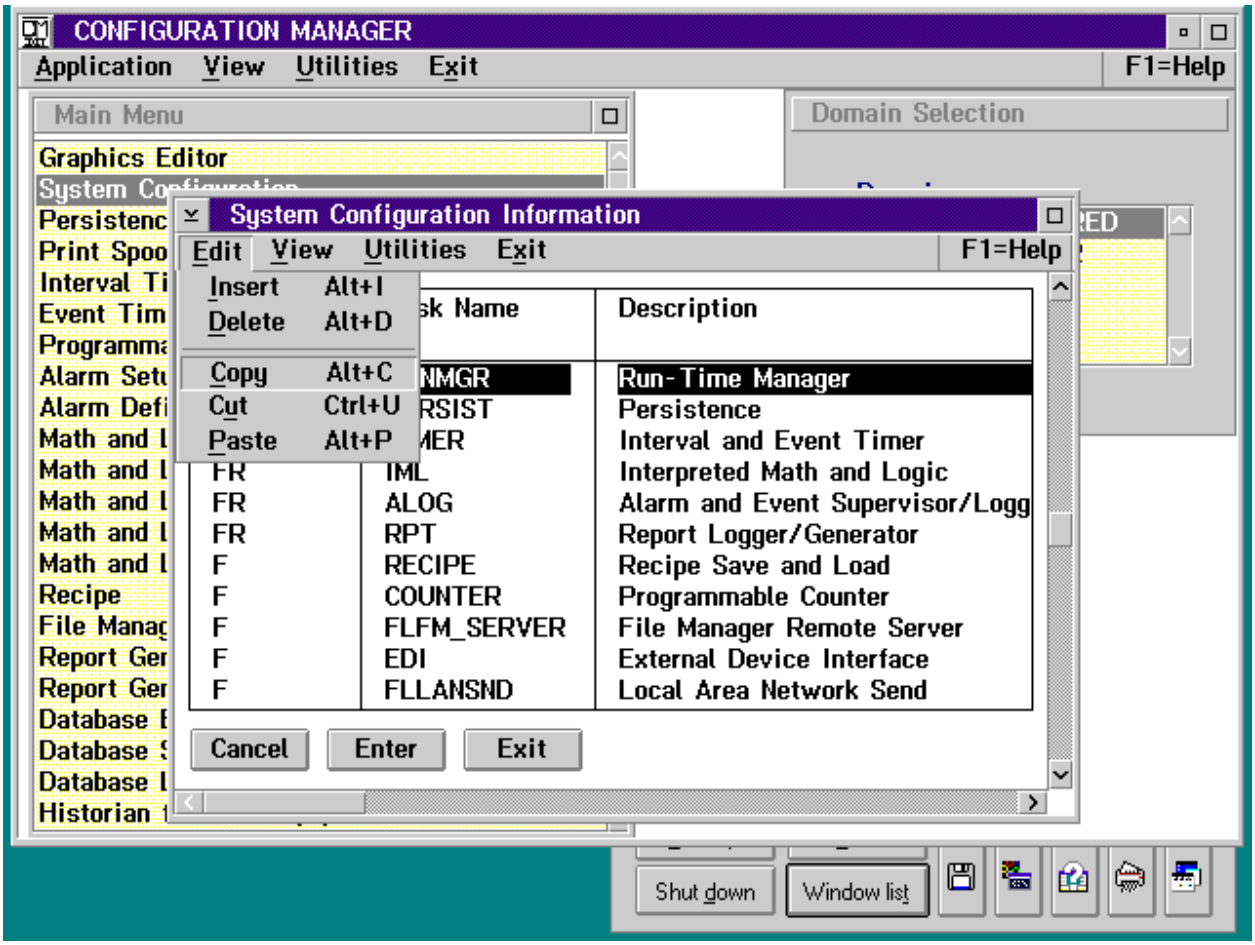


**Figure 2-2: System Configuration Information screen with Copy function highlighted.**

Change the pasted duplicate row according to the following flags:

Flags **S** **Session flag:** Provides a separate window for the task to which all information is directed which can then be called up may require.

**F** **Foreground flag:** Allows task to be displayed in the foreground of the Run-Time Manager window.

**R** **Run flag:** Automatically starts the driver when the system starts. Set the flag to "R" if you do not want a driver screen.

**To get a driver screen, set flags "S" and "F".** The driver screen displays the following information:

- polled station,
- responding station,
- system time,

- driver errors and
- communication status.

**Fill in tag names as follows:**

| | |
|---|---|
| Task Name | ROC ***x*** |
| Description | ROC ***x*** 364 Driver |
| Start Trigger | TASKSTART_S[1+***y***] |
| Task Status | TASKSTATUS_S[1+***y***] |
| Task Message | TASKMESSAGE_S[1+***y***] |
| Display Status | TASKDSTATUS_S[1+***y***] |
| Display Name | TASKNAME_S[1+***y***] |
| Display Descriptor | TASKDESC_S[1+***y***] |

where ***x*** = driver number (1 up to 6)
where ***y*** = current number in tag array

| | | |
|---|---|---|
| Start Order | ***N*** | if installing the first task |
| | ***N+1*** | if installing the second task, etc. |

where ***N =*** a number larger than any currently installed task that may download data to the ROC

**Start Order prioritizes which command lines activate first, starting from 1.** This prioritization is very important as some command lines depend on others in order to work well.

| | |
|---|---|
| Priority | 201 |
| Executable File | BIN/ROC |
| Application Directory | blank |
| Program Directory | blank |
| Program Arguments | ***N*** or ***n*** for **No** printer spooling of driver information. Any other entry, including a blank entry, will spool driver information to the printer. |

**You should start typical tasks like Alarm Manager and Batch Manager *before* starting the ROC driver.** Starting these typical tasks first eliminates the chance of downloading

default values from the ROC. If you do download the ROC default values, you must reconfigure the ROC all over again.

# Section 3

# CONFIGURING THE ROC TABLES

# 3. Configuring the ROC Tables

## 3.1 Introduction to ROC Table Configuration

**We use 3 tables to configure the ROC driver.** You access them by clicking on the ROC Driver you want to configure in the ***Main Menu***.

**The *ROC Driver Communication Information* screen is a single line display**. It configures the general type of information needed by the driver.

**The *ROC Station Control* screen consists of one line of information for each implemented station.** It contains a specific station's configuration information. You define items such as the communications status tag on this screen.

**You define specific point information for a particular station on the *ROC Station Read/Write Information* screen.** There is one ***Read/Write*** screen for each implemented station. Each implemented point for the station needs a separate line on the ***Read/Write*** screen. Explained below is the configuration information you need.

The TAG NAME entry may have up to 16 characters.

**WARNING!!**

**To store the ROC data correctly, each ROC data type maps to one or more USDATA type points.** For example, the USDATA type FLOAT maps to Engineering Units (EU). You must observe this mapping restriction for all points when you build your database. **Section 3.4** lists the matching types.

**While initializing, the driver builds internal tables from the screen information.** The driver checks each entered tag for adherence to the type restriction. All violations will appear on the driver screen. The driver will not start until you fix each violation. You fix violations by redefining the tag type to the USDATA type. Check that the ROC tag type correctly matches the parameter definition of the type point.

To print out a copy of these errors, remove ´**n**´ from ***Program Arguments*** in the ***System Configuration Information*** screen.

## 3.2 ROC Driver Communication Information Screen

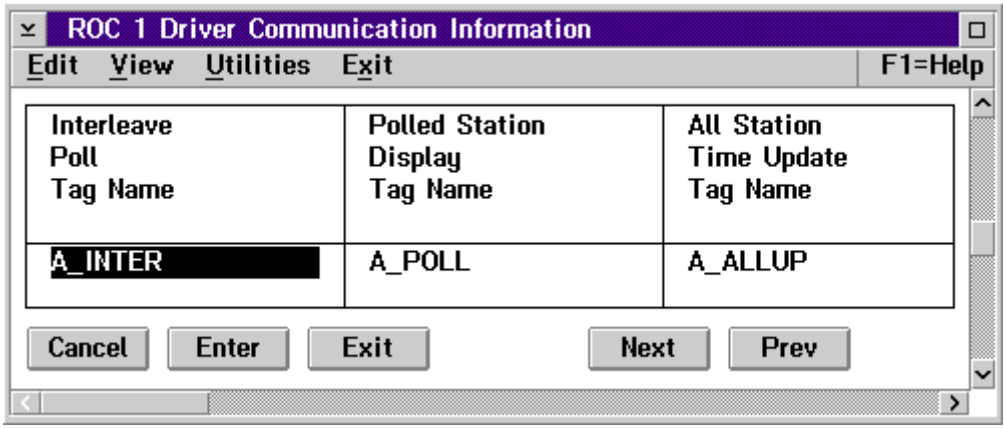


**Figure 3-1: ROC Driver Communication Information screen.**

**This screen defines the driver common information in only one line.**. See below for more information on each item.

### Interleave Poll Tag Name

**This tag identifies the requested interleave station number.** The driver checks this tag location before each poll. The driver polls the station number stored there when it is between each regular poll request. We know it as ***interleave mode***.

Interleave mode continues until it has made a total of 30 interleave poll requests. If you set the station to "**0**" (zero), it returns to "round robin" polling.

**You can start interleave polling in one of 2 ways:**

1. Send a non-data request command to any station, automatically starting interleave on that station; or
2. Use any USDATA function external to the driver by setting the interleave scan tag to the desired station.

Set the interleave tag to "**0**" to cancel it. If you set a new interleave station while interleaving, you cancel the current interleave station and replace it with the new station. As before, the new station will interleave for 30 polls.

### Polled Station Display Tag Name

**This tag shows which station the driver polls.** The driver updates the tag before each poll request.

## All Station Time Update

**This tag requests an all-station time update.** Before each poll, the driver checks this tag. If it detects a non-zero value, the driver sends the Master Time to all stations simultaneously.

## Serial Port Name

**This tag names the serial port for communication use.** Its name (9 characters maximum) must match the serial port name expected by the system. e.g. If using communications port 2, the entry should be "COM2".

## Baud Rate

**Enter the value of the desired baud rate here.** Use the **< ALT-/ >** key combination to choose from the table of valid rates. Valid speeds for ROC communication are 300, 600, 1200, 2400 and 9600 baud.

## Response Time-out (0.1 seconds)

**This tag lists the set time limit in one-tenth second increments for a response.** If the driver receives no response after the set limit, it displays "NO". Response time means the time between transmission of the last poll character and the time the station receives the first character. e.g. For a 2 second wait time, enter 20.

## Normal Retry Count

**This tag shows the number of times the driver can re-poll the station before setting its status to "FAIL".**

## Fail Retry Count

**This tag determines the number of times to poll a station showing a "FAIL" condition.** The normal value is "**1**" to minimize the time spent trying to talk to a known failed station.

## Inter (Dial) Station Delay (0.1 seconds)

**This *delay time* serves 2 purposes, depending on the station's operation mode.**

***Polled mode:*** The driver delays transmission for the length of time you set. You enter the time in one-tenth second increments i.e. enter 20 for a 2 second wait time. The set time limit can differ for each poll.

***Dial out mode:*** The driver waits up to this time limit for Data Carrier Detector (DCD) after dialing the telephone number. Enter 20 to get the typical value for this delay of 2 seconds.

## Same Station Delay (0.1 seconds)

**The driver delays transmission between multiple polls on the same station.** Enter the time in one-tenth second increments to determine the length of the delay. e.g. Enter 20 to produce a 2 second wait time.

## Default Configuration

**The driver operates in four modes: Polled, Monitor, Unsolicited and Dial Up.** Below we describe each mode.

**Polled = 1**

***POLLED mode:*** The driver makes poll requests in a continuous first-to-last poll called a "round robin". Only the interleave and interstation delay times can change "round robin" polling. The driver sends out commands when it detects that the user has changed a value or asked for an action.

**Monitor = 0**

***MONITOR mode:*** The driver continuously monitors the received data line and processes the data it gets. You would only set your driver into this mode if you wanted it in a standby or backup configuration.

**Unsolicited = 2 (UNSOL)**

***UNSOL mode:*** The driver continuously monitors the received data line for a valid station upset request. This mode reports to the driver the exceptions to normal operation. Once the driver receives a message like this, it immediately polls the identified station to get that station's data.

**If the Station telephone number is in the ROC station control table, the driver assumes communication is by telephone.** In

this case, the ROC calls the host machine via the telephone line, connects, and then allows the ROC to solicit a regular and full poll. The ROC Station Control Table provides the regular poll tag allowing periodic station polls. The driver sends out commands when it detects a changed value.

**Dial Out = 3 (DIAL)**

***Dial Out mode:*** Dial Out mode replaces the continuous "round robin" polling with a single poll of all active stations. The driver automatically Dials Out to connect with each station before each poll. Once the poll finishes, the driver suspends itself until it detects a change in the regular poll tag. The driver sends out commands when it detects a changed value.

**For the Dial Out mode to work well, you must meet these requirements:**

- you must use a Hayes compatible modem;
- you must configure the modem to disconnect if it loses Data Terminal Ready (DTR) and supply Data Carrier Detector (DCD) when connected.

You may choose the mode from those available by using the **< ALT-/ >** key combination.

## Transmit Turn On Delay (0.01 seconds)

**After Requesting To Send (RTS), the driver delays for the set amount of time before sending character transmissions.** Enter the time in one hundredth second increments. e.g. To produce a 0.2 second wait time, enter 20. In the case of the Dial Out configuration, the time delay comes after the driver receives the DCD.

**NOTE: The driver has hard coded a number of the more obscure communication channel parameters to minimize the configuration required.** However, you must make sure your ROC and FL values match and meet each other´s parameters or you will have a breakdown in communication. **The hard coded parameters are as follows:**

Number of data bits = **8**
Number of stop bits = **1**
Parity = (none)

## Global Regular Tag Name

**The Global Regular Tag Name tells the driver it needs to start a regular poll request when it is in the Unsolicited and Dial Out operation modes.** The driver checks this tag regularly. If it is set, the driver sets all active regular poll tags. Also, you can use this tag to set individual poll tags.

### Global Full Tag Name

**This tag performs the same function as the regular poll above except it relates to the full update request.**

### Global History Tag Name

**This tag performs the same function as the regular poll above except it relates to the historical data update request**.

### Configuration Tag Name (Analog)

**When you create a tag a screen pops up and asks you to define the tag (if you want to redefine the tag later press < Ctrl-t >).** That pop-up screen also asks you to define the default value. When the Driver starts, the Default Configuration Tag Name value becomes the Configuration Tag Name value. This tag lets you change the driver's mode. Change the mode by inputting a value between 0 and 3 inclusive, into the configuration tag name. If you enter a value less than 0 or greater than 3, nothing happens.

## 3.3 ROC Station Control screen

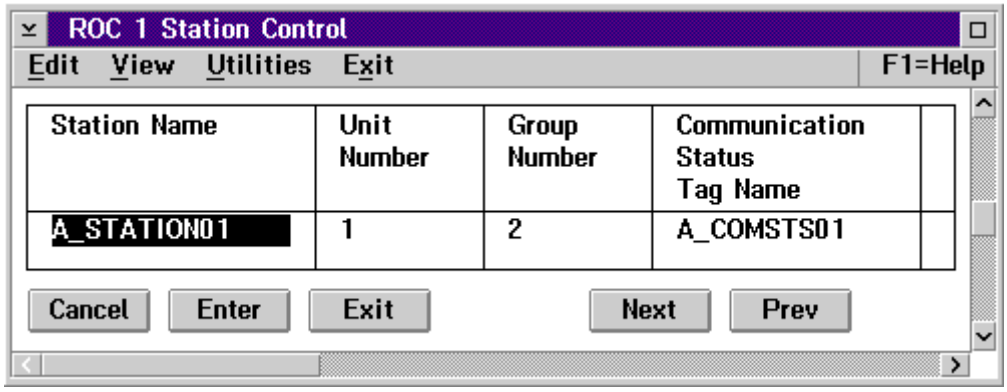


**Figure 3-2: ROC Station Control screen.**

**The ROC Station Control screen defines parameters common to all individual stations.** The drive polls the stations in numerical order, beginning at station 1. The driver reserves station 0 for future "all station" applications. The interleave and polled station tags use the station number as defined by this list. Do not confuse this Station Number with the station's ROC ***Unit*** Number. For example, the station number is 2 if it is second on the list. The actual unit number for that station might differ completely, depending on its definition (see Unit Number below). On-screen, the parameters are listed horizontally, spreadsheet style. We discuss each parameter below.

## Station Name

**This 16 character name identifies the station and its associated table as defined in Section 3.4. You can name it as you wish.**

## Unit Number

**This value identifies the station targeted by the poll.** It is part of the transmit message. The Unit Number should match the "***ROC Address***" as shown on the ROC System Variables Display (GV101 Panel).

## Group Number

**This value identifies the group number of the remote station targeted by the poll.** It is part of the transmit message of the poll. The Group number should match the "ROC Group" as shown on the ROC System Variables Display (GV101 Panel).

## Communications Status Flag

**This tag identifies the storage location for the station's communication status.** The driver sets this value to "**0**" if communications to the targeted station are NORMAL. The driver sets this value to "**1**" if communications to the targeted station have failed.

## Restart Status Tag Name (Digital)

**This tag detects an RTU restart.** The driver writes "**1**" to the restart status tag name. The host time automatically writes to the RTU. The Restart tag resets to "**0**".

## Poll Count Tag Name

**The driver adds 1 to this tag value each time the driver makes a poll request to the associated station.** This count increases incrementally to approximately 2.147 * $10^9$ (4 bytes). If you need this value to keep daily and/or hourly counts, you must use a separate program to reset the values to zero, among other possibilities.

## Incomplete Count Tag Name

**The driver adds 1 to this tag each time the driver detects an Incomplete Response to a poll message.** An incomplete message receives one or more characters but failed to finish the transmission in the allotted time (see response Time Out value). This count increases incrementally to approximately 2.147 * $10^9$ (4 bytes). If you need this value to keep daily and/or hourly counts, you must use a separate program to reset the values to zero, among other possibilities.

## No Response Count Tag Name

**The driver adds 1 this tag each time the driver gets NO RESPONSE.** A NO RESPONSE defines itself as a poll message that received no characters in the allotted time. This count increases incrementally to approximately 2.147 * $10^9$ (4 bytes). If you need this value to keep daily and/or hourly counts, you must use a separate program to reset the values to zero, among other possibilities.

## Poll Enable Tag Name

**This tag's value lets functions outside the driver start or stop polling individual stations.** The driver checks the status of this tag before polling. If the tag value is not zero, the driver will not poll the station. This feature removes stations from the regular poll list for station servicing or other reasons.

## Time Update Tag Name

**This tag requests a time update for an individual station.** The driver checks this tag before it polls the target station. If the driver finds a non-zero value, it sends the Master Time to the station. The driver automatically sets this tag value if it detects a station restart status in the message returned from the station. In addition, functions outside the driver may set this value any time it needs to download the Master Time.

Normally, the driver downloads the Master Time once per day to ensure system-wide time synchronization.

## Regular Update Tag Name

**Use this tag to request a regular data poll.** The driver uses this flag in the unsolicited mode to provide a background data poll. This poll ensures data integrity. In the Dial Out mode, it scans the station.

Normally, the driver polls each station for current data using the ROC function codes, **0** and **167**.

**These function codes return the following information:**

- system analog inputs
- discrete inputs
- timed inputs
- field analog inputs
- gas (AGA) flow rate
- gas (AGA) energy rate
- gas (AGA) volume
- gas (AGA) accumulated energy
- raw pulse accumulator
- pulse input rate
- accumulated pulse input
- loop (PID) status
- loop (PID) primary setpoint
- loop (PID) secondary setpoint
- tank accumulation
- current analog output value
- current timed output value
- current discrete output value
- soft point values
- current date and time

**You must ask specifically for information not in the above list.** You do this by requesting a full update and historical and soft point request tags. After a regular data poll, the driver checks the 3 tags. If set, they trigger the appropriate poll and update the corresponding definitions. Once polled, the flag resets to prevent multiple requests.

## Full Update Request Tag Name

**This tag requests a full station update.** A Full Update Request polls the following information:

- all parameters for each implemented DIX type point (Table 22 on page 26 of the ROC manual);
- all parameters for each implemented DOX type point (Table 23 on page 27 of the ROC manual);
- all parameters for each implemented AIX type point (Table 24 on page 27 of the ROC manual);
- all parameters for each implemented AOX type point (Table 25 on page 28 of the ROC manual);
- all parameters for each implemented PIX type point (Table 26 on page 28 of the ROC manual);
- all parameters for each implemented PID type point (Table 27 on page 29 of the ROC manual);
- all parameters for each implemented AGA type point (Table 28 on page 29 of the ROC manual);
- all parameters for each implemented AGAX type point (Table 30 on page 31 of the ROC manual);
- all parameters for each implemented FST type point (Table 26 on page 34 of the ROC manual);
- all the above poll requests made on a per point basis for all parameters.

**The received data only updates implemented point parameters.**

## Historical Update Tag Name

**This tag requests an update of all historical type data points defined for the station.** In this example, the tag value targets specific data for the poll.

- Use a number between 1 and 31 as the current month's day number.
- Use a number between -1 and -31 as the previous month's day number.

### AGA Parameter Change Tag Name

**This parameter marks changes to the AGA values.** This tag sets to a "1" if the following configured AGA parameters change:

| | | | |
|---|---|---|---|
| 0 | 14 | 44 | 50 |
| 3 | 15 | 45 | 51 |
| 4 | 16 | 47 | 52 |
| 7 | 22 | 48 | 53 |
| 8 | 43 | 49 | |

### Manual Command Tag Name

**Normally, the driver checks all incoming commands for all stations after each station poll finishes.** In large systems, the checking process may take several seconds. Specify a tag name in this column to reduce the delay. When you type a tag into this column, the driver will process only an incoming station's commands when it is a non-zero tag value. A non-zero tag value setting tells the driver to process the incoming commands for that station. It then re-sets the tag value to zero.

**NOTE: When using a tag in this column, you must set the tag you need.** Normally you would need a "SEND DATA" button.

### Dial Out Character String

**This slot defines the station telephone number to use in the Dial Out mode.** The telephone number can have 80 characters. The string must begin with the "**AT**" Hayes compatible command or the modem will ignore it.

## 3.4 ROC Station Read/Write Information

**See Figure 3-3 on following page.**

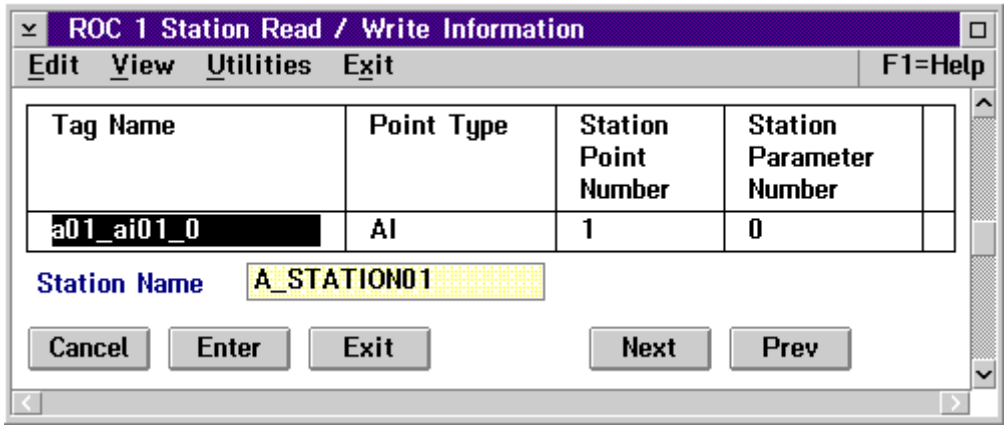


**Figure 3-3: ROC Station Read/Write Information**

**Each implemented station needs one screen.** Each implemented point for a specific station has a separate input line on this screen. Below, we describe each input line.

## Tag Name

**This entry contains the tag name for data storage.** The tag may be either an existing tag previously defined or a new tag created to store data. In either case the USDATA tag type must correspond to the ROC type definition described below.

## Point Type

**You must identify the ROC information as a particular type so the driver can properly process it.** Below are the acceptable point types and a definition of each type. Select this list when configuring a point by using the < **ALT-/** > key combination.

**DI**

**This type defines a status input point.** This data returns as part of a regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| DIGITAL | 0 | current status |

**AI**

**This type defines an Analog Input point.** This data returns as part of a regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | current value (eng units) |

**Parameter "0" defines field analog values**. Parameters **1 to 5** implement the special system analogs as follows:

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 1 | TX voltage (eng units) |
| FLOAT | 2 | input voltage (eng units) |
| FLOAT | 3 | AUX 2 voltage (eng units) |
| FLOAT | 4 | AUX 1 voltage (eng units) |
| FLOAT | 5 | board temp (eng units) |

**NOTE**

**You can individually implement the above parameters as desired.**

**TI**

**This type defines a digital Timed Input point.** This data returns as part of a regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | timed value (eng units) |

**AGAF**

**This type defines an AGA Flow type point.** This data returns as part of a regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | current gas flow (MCF/day) |
| FLOAT | 1 | current energy flow (MMBTU/day) |

| | | |
|---|---|---|
| FLOAT | 2 | total volume since contract hour (MCF) |
| FLOAT | 3 | total energy since contract hour (MMBTU) |

**ACC**

**This type defines a pulse input type point.** This data returns as part of a regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| LONGANA | 0 | raw accumulator counts |
| FLOAT | 1 | rate / unit time (eng units) |
| FLOAT | 2 | total today (eng units) |

**PIDF**

**This type defines an abbreviated PID loop point.** Note that you must use this type to define loop information when you want a regular update (as opposed to a full update).

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| ANALOG | 0 | loop status |
| FLOAT | 1 | primary setpoint |
| FLOAT | 2 | secondary setpoint |

**TANK**

**This type defines a Tank type point.** This data returns as part of a regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | volume since contract hour |

**DO**

**This type defines a Digital control point.** The driver updates the status value into the tag slot from the poll it received. The driver also uses this table information to send commands.

| USDATA | Valid |
|---|---|

| Type | Parameter | Description |
|---|---|---|
| DIGITAL | 0 | current state |

**AO**

**This type defines an Analog setpoint Output point.** The driver uses this information to send the setpoint to the station. It also updates the current value as received from the station into the tag slot.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | current value (eng units) |

**TO**

**This type defines a Timed discrete Output point.** The driver uses this table information to send the value to the station. The driver also updates the tag slot with the current value as received from the station's response to the regular poll request.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | current value (eng units) |

**PID**

**This type defines a PID control block point.** The driver uses this table information to download parameters to the station. The driver updates the parameter value sent by the station into the tag slot. Each time a PID downloads the full update is set for that station.

| USDATA Type | Parameter | Valid Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| ANALOG | 1 | yes | control type |
| ANALOG | 2 | no | switch status |
| LONGANA | 3 | no | actual scan time |
| LONGANA | 4 | yes | PRI input point |
| LONGANA | 5 | yes | PRI output point |

| | | | |
|---|---|---|---|
| FLOAT | 6 | yes | PRI SW setpoint |
| LONGANA | 7 | yes | PRI switch PV |
| MESSAGE | 8 | yes | PRI switch mode |
| LONGANA | 9 | yes | OVR input point |
| LONGANA | 10 | yes | OVR output point |
| FLOAT | 11 | yes | OVR SW setpoint |
| LONGANA | 12 | yes | OVR switch PV |
| MESSAGE | 13 | yes | OVR switch mode |
| FLOAT | 14 | yes | setpoint |
| FLOAT | 15 | yes | setpoint change/min |
| LONGANA | 16 | yes | loop period |
| FLOAT | 17 | yes | proportional gain |
| FLOAT | 18 | yes | integral gain |
| FLOAT | 19 | yes | derivative gain |
| FLOAT | 20 | yes | scale factor |
| FLOAT | 21 | yes | integral deadband |
| FLOAT | 22 | no | process variable |
| FLOAT | 23 | yes | output |
| FLOAT | 24 | no | switch PV |
| LONGANA | 25 | yes | min control time |
| FLOAT | 26 | yes | setpoint |
| FLOAT | 27 | yes | setpoint change/min |
| LONGANA | 28 | yes | loop period |
| FLOAT | 29 | yes | proportional gain |
| FLOAT | 30 | yes | integral gain |
| FLOAT | 31 | yes | derivative gain |
| FLOAT | 32 | yes | scale factor |
| FLOAT | 33 | yes | integral deadband |
| FLOAT | 34 | no | process variable |
| FLOAT | 35 | yes | output |
| FLOAT | 36 | no | switch PV |

Please refer to the Fisher ROC 364 documentation for a complete definition of parameters.

**AGA**

**This type defines an AGA control block point.** The driver uses this table information to download parameters to the station. The current parameter value sent by the station updates into the tag slot as part of a full update request. For this reason, each time an AGA downloads the full update sets for that station.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| FLOAT | 1 | yes | latitude |
| FLOAT | 2 | yes | elevation |
| ANALOG | 3 | yes | calculation method |
| ANALOG | 4 | yes | options |
| FLOAT | 5 | yes | specific gravity |
| FLOAT | 6 | yes | heating value |
| FLOAT | 7 | yes | grav accel corr |
| LONGANA | 8 | yes | scan period |
| FLOAT | 9 | yes | pipe diameter |
| FLOAT | 10 | yes | orifice diameter |
| FLOAT | 11 | yes | orifice MSRD temp |
| ANALOG | 12 | yes | orifice material |
| MESSAGE | 13 | yes | description (30) |
| ANALOG | 14 | yes | alarm code |
| FLOAT | 15 | yes | low alarm |
| FLOAT | 16 | yes | high alarm |
| FLOAT | 17 | yes | viscosity |
| FLOAT | 18 | yes | specific heat ratio |
| FLOAT | 19 | yes | contract pressure |
| FLOAT | 20 | yes | contract temp |
| FLOAT | 21 | yes | diff PR low cutoff |
| FLOAT | 22 | yes | grav correction |
| FLOAT | 23 | yes | nitrogen |
| FLOAT | 24 | yes | carbon dioxide |
| FLOAT | 25 | yes | hydrogen sulfide |
| FLOAT | 26 | yes | water |
| FLOAT | 27 | yes | helium |
| FLOAT | 28 | yes | methane |
| FLOAT | 29 | yes | ethane |
| FLOAT | 30 | yes | Propane |
| FLOAT | 31 | yes | n-butane |
| FLOAT | 32 | yes | i-butane |
| FLOAT | 33 | yes | n-pentane |
| FLOAT | 34 | yes | i-pentane |
| FLOAT | 35 | yes | n-hexane |
| FLOAT | 36 | yes | n-heptane |
| FLOAT | 37 | yes | n-octane |
| FLOAT | 38 | yes | n-nonane |
| FLOAT | 39 | yes | n-decane |
| FLOAT | 40 | yes | oxygen |
| FLOAT | 41 | yes | carbon monoxide |
| FLOAT | 42 | yes | hydrogen |
| ANALOG | 43 | no | spare |
| ANALOG | 44 | yes | enable stacked DP |

| | | | |
|---|---|---|---|
| LONGANA | 45 | yes | low diff PR number |
| LONGANA | 46 | yes | diff PR number |
| LONGANA | 47 | yes | static PR number |
| LONGANA | 48 | yes | temperature number |
| FLOAT | 49 | yes | low diff PR setpoint |
| FLOAT | 50 | yes | high DP setpoint |
| FLOAT | 51 | yes | HW |
| FLOAT | 52 | yes | PF |
| FLOAT | 53 | yes | TF |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**DIX**

**This type defines an auxiliary status input type point.** The driver uses this type for status points where it needs more information on the point. It also uses this table information to download parameters to the station. The current parameter value as received from the station also updates into the tag slot. For this reason, each time a DIX downloads, the full update sets for that station.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| ANALOG | 1 | yes | filter |
| ANALOG | 2 | yes | status |
| ANALOG | 3 | yes | modes |
| ANALOG | 4 | no | alarm code |
| LONGANA | 5 | yes | accum value |
| LONGANA | 6 | yes | on counter |
| LONGANA | 7 | yes | off counter |
| LONGANA | 8 | yes | 0% count |
| LONGANA | 9 | yes | 100% count |
| LONGANA | 10 | yes | max count |
| MESSAGE | 11 | yes | units |
| LONGANA | 12 | yes | scan period |
| FLOAT | 13 | yes | zero eng unit |
| FLOAT | 14 | yes | span eng unit |
| FLOAT | 15 | yes | low alarm |
| FLOAT | 16 | yes | high alarm |
| FLOAT | 17 | yes | lolo alarm |
| FLOAT | 18 | yes | hihi alarm |
| FLOAT | 19 | yes | delta alarm |
| FLOAT | 20 | yes | alarm deadband |

| | | | |
|---|---|---|---|
| FLOAT | 21 | yes | eng unit value |
| LONGANA | 22 | no | TDI count |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**DOX**

**This type defines an auxiliary status output point.** The driver uses this type for status output points where it needs more information on the point. It also uses this table information to download parameters to the station. The driver updates the current parameter from the poll request into the tag slot. For this reason, the full update sets each time a DOX downloads.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| LONGANA | 1 | yes | time on |
| ANALOG | 2 | no | spare |
| ANALOG | 3 | yes | status |
| ANALOG | 4 | yes | mode |
| ANALOG | 5 | no | alarm code |
| LONGANA | 6 | yes | accum value |
| MESSAGE | 7 | yes | units (10) |
| LONGANA | 8 | yes | cycle time |
| LONGANA | 9 | yes | 0% count |
| LONGANA | 10 | yes | 100% count |
| FLOAT | 11 | yes | low reading eng unit |
| FLOAT | 12 | yes | hi reading eng unit |
| FLOAT | 13 | yes | eng unit value |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**AIX**

**This type defines an auxiliary Analog Input type point.** The driver uses this type when it needs more information on the point. It also uses the table information to download parameters to the station. The driver updates the current parameter values as received from the full poll of the station into the tag slot. For this reason, the full update is set for that station each time it performs an AIX download.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| MESSAGE | 1 | yes | units (10) |
| LONGANA | 2 | yes | scan period |
| LONGANA | 3 | yes | filter |
| LONGANA | 4 | yes | adj a/d 0% |
| LONGANA | 5 | yes | adj a/d 100% |
| FLOAT | 6 | yes | low reading eng unit |
| FLOAT | 7 | yes | hi reading eng unit |
| FLOAT | 8 | yes | low alarm eng unit |
| FLOAT | 9 | yes | hi alarm eng unit |
| FLOAT | 10 | yes | lolo alarm eng unit |
| FLOAT | 11 | yes | hihi alarm eng unit |
| FLOAT | 12 | yes | delta alarm eng unit |
| FLOAT | 13 | yes | alarm deadband |
| FLOAT | 14 | yes | filtered eng unit |
| ANALOG | 15 | yes | mode |
| ANALOG | 16 | no | alarm code |
| LONGANA | 17 | no | raw A/D input |
| LONGANA | 18 | no | actual scan |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**AOX**

**This type defines an auxiliary Analog Output type point.** The driver uses this type when it needs more information on the point. The driver also uses this table information to download parameters to the station. It updates the current parameter values (from the station's full update), into the tag slot. For this reason, the full update is set for that station each time it performs an AOX download.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag id (10) |
| MESSAGE | 1 | yes | units (10) |
| LONGANA | 2 | yes | adj a/d 0% |
| LONGANA | 3 | yes | adj a/d 100% |
| FLOAT | 4 | yes | low reading eng unit |
| FLOAT | 5 | yes | hi reading eng unit |
| FLOAT | 6 | yes | eng unit value |
| ANALOG | 7 | yes | mode |
| ANALOG | 8 | no | alarm code |
| LONGANA | 9 | no | raw a/d output |

Please refer to the Fisher ROC 364 documentation for a complete definition of parameters.

**PIX**

**This type defines an auxiliary Pulse Input type point.** The driver uses this type for pulse inputs when it needs more information on the point. It also uses the table information to download parameters to the station. The station receives the current parameter from a full update request and uploads it into the tag slot. For this reason, the full update is set for that station each time a PIX downloads.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| MESSAGE | 1 | yes | units (10) |
| ANALOG | 2 | yes | rate flag |
| ANALOG | 3 | yes | rate period |
| ANALOG | 4 | no | type |
| LONGANA | 5 | yes | scan period |
| FLOAT | 6 | yes | conversion |
| FLOAT | 7 | yes | low alarm eng unit |
| FLOAT | 8 | yes | hi alarm eng unit |
| FLOAT | 9 | yes | lolo alarm eng unit |
| FLOAT | 10 | yes | hihi alarm eng unit |
| FLOAT | 11 | yes | delta alarm eng unit |
| FLOAT | 12 | yes | alarm deadband |
| FLOAT | 13 | yes | eng unit value |
| ANALOG | 14 | yes | mode |
| ANALOG | 15 | no | alarm code |
| LONGANA | 16 | yes | accum value |
| FLOAT | 17 | no | current rate |
| FLOAT | 18 | yes | today's total |
| FLOAT | 19 | no | yesterday's total |

Please refer to the Fisher ROC 364 documentation for a complete definition of parameters.

**FST**

**This type defines Funnction Sequence Table type points.** The driver also uses this table information to download parameters to the station. The current parameter value (received as part of a full update request) updates

the tag slot. For this reason, the full update is set each time an FST downloads.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID (10) |
| FLOAT | 1 | yes | result register |
| FLOAT | 2 | yes | register # 1 |
| FLOAT | 3 | yes | register # 2 |
| FLOAT | 4 | yes | register # 3 |
| FLOAT | 5 | yes | register # 4 |
| FLOAT | 6 | yes | register # 5 |
| FLOAT | 7 | yes | register # 6 |
| FLOAT | 8 | yes | register # 7 |
| FLOAT | 9 | yes | register # 8 |
| FLOAT | 10 | yes | register # 9 |
| FLOAT | 11 | yes | register # 10 |
| LONGANA | 12 | yes | timer # 1 |
| LONGANA | 13 | yes | timer # 2 |
| LONGANA | 14 | yes | timer # 3 |
| LONGANA | 15 | yes | timer # 4 |
| MESSAGE | 16 | yes | message # 1 (30) |
| MESSAGE | 17 | yes | message # 2 (30) |
| MESSAGE | 18 | no | message data (10) |
| ANALOG | 19 | yes | misc |
| ANALOG | 20 | yes | misc |
| ANALOG | 21 | yes | misc |
| ANALOG | 22 | yes | misc |
| ANALOG | 23 | yes | compare flag SVD |
| ANALOG | 24 | yes | run flag |
| LONGANA | 25 | yes | code size |
| LONGANA | 26 | yes | inst pointer |
| LONGANA | 27 | yes | execution delay |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**AGAX**

**This type defines more AGA loop parameters.** A full update request updates any AGAX type points.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | hw |
| FLOAT | 1 | pf |
| FLOAT | 2 | Tf |
| FLOAT | 3 | MCF/day |
| FLOAT | 4 | MMBTU/day |
| FLOAT | 5 | MCFs today |
| FLOAT | 6 | MMBTUs today |
| FLOAT | 7 | MCFs yesterday |
| FLOAT | 8 | MMBTUs yesterday |
| FLOAT | 9 | Pressure extension |
| FLOAT | 10 | C prime |
| FLOAT | 11 | sample time |
| FLOAT | 12 | expansion factor |
| FLOAT | 13 | Fr |
| FLOAT | 14 | Ftf |
| FLOAT | 15 | Fpv |
| FLOAT | 16 | Fgr |
| FLOAT | 17 | Fb |
| FLOAT | 18 | Fpb |
| FLOAT | 19 | Ftb |
| FLOAT | 20 | Fa |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**HIST**

**This type defines historical type points.** The parameters match the ROC data returned with an opcode 128 request. They start by setting the historical update request flag for that station.

| USDATA Type | Valid Parameter | Description |
|---|---|---|
| FLOAT | 0 | hour 0 |
| FLOAT | 1 | hour 1 |
| FLOAT | 2 | hour 2 |
| FLOAT | 3 | hour 3 |
| FLOAT | 4 | hour 4 |
| FLOAT | 5 | hour 5 |
| FLOAT | 6 | hour 6 |

| | | |
|---|---|---|
| FLOAT | 7 | hour 7 |
| FLOAT | 8 | hour 8 |
| FLOAT | 9 | hour 9 |
| FLOAT | 10 | hour 10 |
| FLOAT | 11 | hour 11 |
| FLOAT | 12 | hour 12 |
| FLOAT | 13 | hour 13 |
| FLOAT | 14 | hour 14 |
| FLOAT | 15 | hour 15 |
| FLOAT | 16 | hour 16 |
| FLOAT | 17 | hour 17 |
| FLOAT | 18 | hour 18 |
| FLOAT | 19 | hour 19 |
| FLOAT | 20 | hour 20 |
| FLOAT | 21 | hour 21 |
| FLOAT | 22 | hour 22 |
| FLOAT | 23 | hour 23 |
| FLOAT | 24 | daily |
| FLOAT | 25 | min |
| FLOAT | 26 | max |
| MESSAGE | 27 | mm-dd-yy |
| ANALOG | 28 | month (1-12) |
| ANALOG | 29 | day (1-31) |

Please refer to Fisher ROC 364 documentation for a complete definition of parameters.

**SOFT**

**This type defines Soft type points.** The parameters correspond to the ROC data returned with an opcode 180 request. They start by setting the Soft Point update request flag for that station. The driver includes any Soft Points found in the read and write tables in the regular poll.

| USDATA Type | Valid Parameter | Down Loadable | Description |
|---|---|---|---|
| MESSAGE | 0 | yes | point tag ID |
| LONGANA | 1 | yes | integer flag |
| FLOAT | 2 | yes | data #1 |
| FLOAT | 3 | yes | data #2 |
| FLOAT | 4 | yes | data #3 |
| FLOAT | 5 | yes | data #4 |
| FLOAT | 6 | yes | data #5 |
| FLOAT | 7 | yes | data #6 |

| | | | |
|---|---|---|---|
| FLOAT | 8 | yes | data #7 |
| FLOAT | 9 | yes | data #8 |
| FLOAT | 10 | yes | data #9 |
| FLOAT | 11 | yes | data #10 |
| FLOAT | 12 | yes | data #11 |
| FLOAT | 13 | yes | data #12 |
| FLOAT | 14 | yes | data #13 |
| FLOAT | 15 | yes | data #14 |
| FLOAT | 16 | yes | data #15 |
| FLOAT | 17 | yes | data #16 |
| FLOAT | 18 | yes | data #17 |
| FLOAT | 19 | yes | data #18 |
| FLOAT | 20 | yes | data #19 |
| FLOAT | 21 | yes | data #20 |

**DATE**

**This point type retrieves date and time from the ROC.**

**NOTE**

This value returns as part of the regular poll.

| USDATA Type | Valid Station | Valid Parameter | Description |
|---|---|---|---|
| MESSAGE | 1 | 0 | DATE<br>e.g.: 08-21-95 = Aug. 21, 1995 |
| MESSAGE | 1 | 1 | TIME<br>e.g.: 11:43:30 |

## Station Point Number

**The Station Point Number is the point number used in the station.** This number links the number used by the station and the USDATA tag.

**NOTE**: When building the database for a particular station, you can define the points in any order.

## Station Parameter Number

**The Station Parameter Number identifies data associated with the station.** It matches the data descriptions provided under the "***Point Type***" heading. When configuring a station database, you only need to define the parameters from which you will draw data. You can define the parameters in any order. You must match each parameter's USDATA type to the parameter given under "***Point Type***". Matching the wrong data type or inputting a parameter number greater than its limit causes an error when the driver starts up.

## MV

**This type is used to define an MVS type point.** This data is returned as part of a regular poll.

| USDATA Type | Valid Parameter | Down Loadable | Description | ROC 407 | 312/364 SI MVS |
|---|---|---|---|---|---|
| float | 0 | yes | Diff. EU | ai16_0/ai20_0/ ai24_0/ai28_0 | *softzz_y |
| float | 1 | yes | Static EU | ai17_0/ai21_0/ ai25_0/ai29_0 | *softzz_y+1 |
| float | 2 | yes | Temperature EU | ai18_0/ai22_0/ ai26_0/ai30_0 | *softzz_y+2 |
| float | 3 | yes | Reverse Diff. EU | ai19_0/ai23_0/ ai27_0/ai31_0 | *softzz_y+3 |

***NOTE:** zz = softpoint selected
y = softpoint parameter

## MVX

**This type is used to define an MVS type point.** This data is currently stored in the host until the new protocol is added.

| USDATA Type | Parameter | Valid Loadable | Down Description | ROC 407 | 312/364 SI MVS |
|---|---|---|---|---|---|
| Message Only | | 0 | No | MVS Point Tag | Host Only Host |
| Analog | 1 | No | MVS Address | Future Use | Future Use |
| Analog Use | 2 | No | MVS Config | Future Use | Future |
| Analog | 3 | No | MVS poll Method | Future Use | Future Use |
| Analog | 4 | No | MVS Spare | Future Use | Future Use |
| Analog | 5 | 312/364 only | MVS Status | Future Use | **softzz_y+4 |
| Analog | 6 | No | MVS Alarms | Future Use | Future Use |
| float | 7 | No | MVS Controller voltage | Future Use | Future Use |
| float | 8 | No | Diff. EU | Future Use | Future Use |
| float | 9 | No | Static EU | Future Use | Future Use |
| float | 10 | No | Temperature EU | Future Use | Future Use |
| float | 11 | No | Reverse Diff. EU | Future Use | Future Use |
| float | 12 | No | Pressure Effect | Future Use | Future Use |
| float | 13 | No | DP Cal. Min | Future Use | Future Use |
| float | 14 | No | DP Cal. Val Point 1 | Future Use | Future Use |
| float | 15 | No | DP Cal. Val Point 2 | Future Use | Future Use |
| float | 16 | No | DP Cal. Val Point 3 | Future Use | Future Use |
| float | 17 | No | DP Cal. Max | Future Use | Future Use |
| float | 18 | No | Static Cal. Min | Future Use | Future Use |
| float | 19 | No | Static Cal. Val Point 1 | Future Use | Future Use |
| float | 20 | No | Static Cal. Val Point 2 | Future Use | Future Use |
| float | 21 | No | Static Cal. Val Point 3 | Future Use | Future Use |
| float | 22 | No | Static Cal. Max | Future Use | Future Use |
| float | 23 | No | TT Cal. Min | Future Use | Future Use |
| float | 24 | No | TT Cal. Val Point 1 | Future Use | Future Use |
| float | 25 | No | TT Cal. Val Point 2 | Future Use | Future Use |
| float | 26 | No | TT Cal. Val Point 3 | Future Use | Future Use |
| float | 27 | No | TT Cal. Max | Future Use | Future Use |
| Analog | 28 | No | Cal Command | Future Use | Future Use |
| Analog | 29 | No | Cal Type | Future Use | Future Use |
| float | 30 | No | Set Valve | Future Use | Future Use |
| float | 31 | No | Manual DP Valve | Future Use | Future Use |
| float | 32 | No | Manual PRS Valve | Future Use | Future Use |
| float | 33 | No | Manual TT Valve | Future Use | Future Use |
| Analog | 34 | No | DP Alarm Mode | Future Use | Future Use |
| Analog | 35 | No | DP Alarm Alarm | Future Use | Future Use |
| float | 36 | No | DP Alarm Low Limit | Host Only | Host Only |
| float | 37 | No | DP Alarm High Limit | Host Only | Host Only |

| | | | | | | |
|---|---|---|---|---|---|---|
| float | 38 | No | DP Alarm Deadband | Future Use | Future Use | |
| float | 39 | No | DP Alarm Fault | Future Use | Future Use | |
| Analog | 40 | No | Static Alarm Mode | Future Use | Future Use | |
| Analog | 41 | No | Static Alarm Alarm | Host Only | Host Only | |
| float | 42 | No | Static Alarm Low Limit | Host Only | Host Only | |
| float | 43 | No | Static Alarm High Limit | Host Only | Host Only | |
| float | 44 | No | Static Alarm Deadband | Future Use | Future Use | |
| float | 45 | No | Static Alarm Fault | Future Use | Future Use | |
| Analog | 46 | No | TT Alarm Mode | Future Use | Future Use | |
| Analog | 47 | No | TT Alarm Alarm | Future Use | Future Use | |
| float | 48 | No | TT Alarm Low Limit | Host Only | Host Only | |
| float | 49 | No | TT Alarm High Limit | Host Only | Host Only | |
| float | 50 | No | TT Alarm Deadband | Future Use | Future Use | |
| float | 51 | No | TT Alarm Fault | Future Use | Future Use | |
| Message | | 52 | No | Diff Point Tag | Host Only | Host Only |
| Message | | 53 | No | Diff Units | Host Only | Host Only |
| float | 54 | No | Diff Low EU Reading | Host Only | Host Only | |
| float | 55 | No | Diff High EU Reading | Host Only | Host Only | |
| Message | | 56 | No | Static Point Tag | Host Only | Host Only |
| Message | | 57 | No | Static Units | Host Only | Host Only |
| float | 58 | No | Static Low EU Reading | Host Only | Host Only | |
| float | 59 | No | Static High EU Reading | Host Only | Host Only | |
| Message | | 60 | No | Temp Point Tag | Host Only | Host Only |
| Message | | 61 | No | Temp Units | Host Only | Host Only |
| float | 62 | No | Temp Low EU Reading | Host Only | Host Only | |
| float | 63 | No | Temp High EU Reading | Host Only | Host Only | |
| Message | | 64 | No | Rev Diff Point Tag | | Host Only Host Only |
| Message | | 65 | No | Rev Diff Units | Host Only | Host Only |
| float | 66 | No | Rev Diff Low EU Reading | Host Only | Host Only | |
| float | 67 | No | Rev Diff High EU Reading | Host Only | Host Only | |
| float | 68 | No | Rev Diff Low Alarm | Host Only | Host Only | |
| float | 69 | No | Rev Diff High Alarm | Host Only | Host Only | |

*__Host Only -__ refers to points currently defined in the host that will be added to the ROC when the new protocol is written.

__**NOTE:__ zz = softpoint selected
y = softpoint parameter

Parameters 0 to 51 are defined based on information received from Fisher.
Parameters 52 to 69 have been added to accommodate Spartan's requirements.

# GLOSSARY

## A

**ACC:** A pulse input point.
**AGA:** An international equation to determine flow.
**AGAF:** AGA Flow point.
**AGAX:** Additional AGA loop parameter point.
**AI:** Analog Input point.
**AIX:** Auxiliary analog input point.
**Analog:** This term describes a device that measures real values.
**AO:** Analog setpoint Output point.
**AOX:** Auxiliary Analog Output point.

## B

**Baud:** The communication speed of the modem in bits per second (bps).

## D

**Data points:** Temporary data storage areas for writing and retrieval.
**DATE:** Point type that retrieves date and time from the ROC.
**DCD:** Data Carrier Detector (modem).
**Dial Out mode:** When the driver tries to communicate with the ROCs.
**DI:** A status input point.
**Discrete:** This term describes an "on" or "off" device. It is also known as a digital device.
**DIX:** Auxiliary status (digital) input point.
**DO:** Digital control Output point.
**DOX:** Auxiliary status (digital) output point.
**Driver:** Also known as the ROC driver.
**DTR:** Data Terminal Ready (modem).

## E

**EU:** Engineering units. Internationally recognized measurements.

## F

**Flag:** An instruction set. Flag types are Foreground, Run and Session.
**FloBoss:** The latest step in efficient flow monitoring, listed as the ROC 407.
**FST:** Function Sequence Table.

# H

**Hard Coded:** A program fixed in a driver. To change it, you must change the program itself.

**Hayes compatible modem:** A modem that uses a well-defined and well-documented command set.

**HIST:** A Historical point.

# I

**Implement:** An item that is active, like a station.

**Increment:** Increasing a value by a set value, usually by one.

**Input:** Gathering information from a device or monitor.

**Interleave:** Alternately polling one station between the others. e.g. Station 1, 2, 1, 3, 1, 4, etc. Station 1 is the interleave station.

**I/O:** Input/Output.

**I/O interface module:** The device driver that lets FactoryLink WINDOWS NT talk to the ROC.

# M

**Master Time:** The driver time that synchronizes with the ROC clocks.

# N

**Non-zero:** A number that is anything but a 0 (zero).

# O

**Opcode:** The communication language abbreviation for Operational Code.

**Output:** Sending instructions and information to a device.

# P

**Parameters:** The rules that configure a system.

**PID:** A PID control block point.

**PIDF:** Abbreviated PID loop poin.

**PIX:** Auxiliary pulse input point.

**Polled mode:** When polls are made continuously in a "round robin".

# R

**RTS:** Request to Send by the Host Modem.
**ROC:** Remote Operator's Control.
**Round Robin polling:** The Host polls each station sequentially and then re-polls each station in the same order.

# S

**Serial Port:** A socket for plugging in peripheral hardware.
**Shared Domain:** An area where both the User and the Host have access.
**Soft point:** A temporary memory spot where you write data in the ROC.
**SOFT:** Soft point.
**Spool:** The PC's preparation to print a file.

# T

**Tag:** A location for data that triggers a reaction from the driver.
**TANK:** A tank input point.
**TI:** Timed Input point.
**TO:** Timed discrete Output point.

# INDEX